

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 08-335211
(43)Date of publication of application : 17.12.1996

(51)Int.Cl. G06F 15/16
G06F 13/00

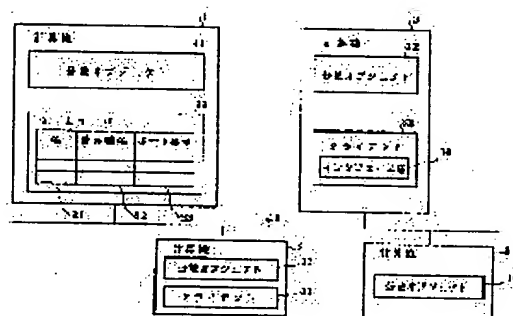
(21)Application number : 07-164684 (71)Applicant : OKI ELECTRIC IND CO LTD
(22)Date of filing : 07.06.1995 (72)Inventor : OTOGAWA SHINICHI

(54) DECENTRALIZED OBJECT ENVIRONMENT EXECUTION SYSTEM

(57)Abstract:

PURPOSE: To facilitate the communication from a client to each decentralized object.

CONSTITUTION: A name server 20 administers the names 22 of computers where respective decentralized objects 11, 12, 13, and 14 are present and port numbers 23 for communications. Further, respective clients 32, etc., are provided with interface parts 30 for communications with the respective decentralized objects. To request one of the decentralized objects to operate data, each client 32, etc., sends the decentralized object name to the name server 20. Consequently, the name server returns information required for a communication with the decentralized object. Therefore, the respective clients 32, etc., need only to be aware of the presence position of the name server 20. Consequently, each client need not pay attention to the presence position of a decentralized object operating on a computer connected to the network 10 and the communication protocol for the decentralized object.



LEGAL STATUS

[Date of request for examination]
[Date of sending the examiner's decision of rejection]
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]
[Date of final disposal for application]
[Patent number]
[Date of registration]
[Number of appeal against examiner's decision of rejection]
[Date of requesting appeal against examiner's decision of rejection]
[Date of extinction of right]

This Page Blank (uspto)

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平8-335211

(43) 公開日 平成8年(1996)12月17日

(51) Int.Cl. ⁶	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 15/16	3 7 0		G 0 6 F 15/16	3 7 0 N
13/00	3 5 7	7368-5E	13/00	3 5 7 Z

審査請求 未請求 請求項の数 3 F D (全 5 頁)

(21) 出願番号 特願平7-164684

(22) 出願日 平成7年(1995)6月7日

(71) 出願人 000000295

沖電気工業株式会社

東京都港区虎ノ門1丁目7番12号

(72) 発明者 乙川 進一

東京都港区虎ノ門1丁目7番12号 沖電気工業株式会社内

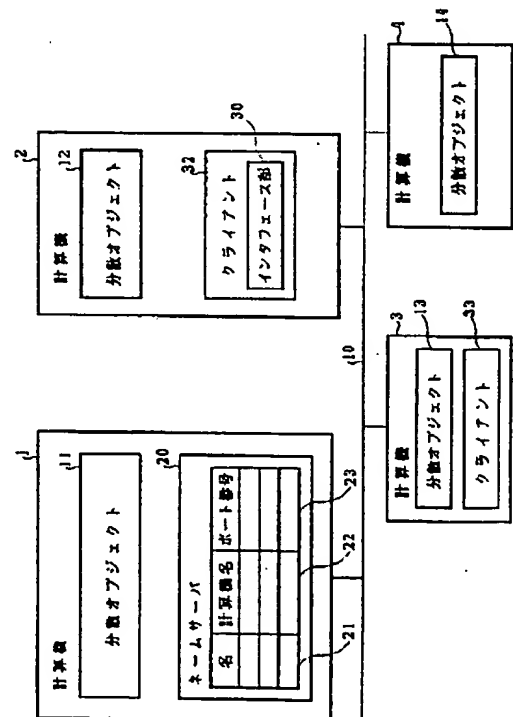
(74) 代理人 弁理士 佐藤 幸男 (外1名)

(54) 【発明の名称】 分散オブジェクト環境実現システム

(57) 【要約】

【目的】 クライアントから各分散オブジェクトへの通信を容易にする。

【構成】 ネームサーバ20によって各分散オブジェクト11、12、13、14の存在する計算機名22、通信のためのポート番号23を管理をしている。また、各分散オブジェクトとの通信のためのインターフェース部30を各クライアント32等に提供している。そして、各クライアント32等がいずれかの分散オブジェクトにデータの操作を依頼する際は、ネームサーバ20に対し、分散オブジェクト名を送信する。これにより、ネームサーバ20がその分散オブジェクトへの通信処理に必要な情報を返信する。従って、各クライアント32等はネームサーバ20の存在位置のみを意識すればよい。この結果、各クライアントは、ネットワーク10に接続された計算機上で動作している分散オブジェクトの存在位置、及びその分散オブジェクトとの通信プロトコルを意識する必要がない。



本発明の分散オブジェクト環境実現システムの一例

【特許請求の範囲】

【請求項 1】 ネットワークに接続された複数の計算機上で動作する分散オブジェクト及び当該分散オブジェクトをサーバとするクライアントの相互間の通信を制御するクライアントサーバシステムにおいて、ネットワークに接続された各計算機上に配置された各分散オブジェクトと、

当該各分散オブジェクトとの通信のためのプロトコルと各分散オブジェクト名とを対応させて管理するネームサーバと、

前記各計算機上に設けられたクライアント内に設けられ、前記ネームサーバに通信先のオブジェクト名を送信するインタフェース部とを備えたことを特徴とする分散オブジェクト環境実現システム。

【請求項 2】 前記各分散オブジェクト内に設けられ、操作の依頼元のクライアント名及び操作内容の履歴を蓄積するヒストリオブジェクトを備えたことを特徴とする請求項 1 記載の分散オブジェクト環境実現システム。

【請求項 3】 前記各分散オブジェクト内に設けられ、操作の直前のデータ内容を保存し、障害時にデータの内容を操作の直前の状態に戻すためのリカバリオブジェクトを備えたことを特徴とする請求項 1 又は請求項 2 記載の分散オブジェクト環境実現システム。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、ネットワークに接続された複数の計算機上で動作する分散オブジェクトが、その計算機上、あるいは他の計算機上で動作する分散オブジェクトと通信するための分散オブジェクト環境実現システムに関するものである。

【0002】

【従来の技術】 一般に、ネットワークで接続された複数の計算機上で、プロセスがメッセージの交換をするとき、オペレーティングシステムによって提供されているメッセージ通信機構を利用する。メッセージ通信の機構には、TCP/IP、UDP/IP のプロトコルが使われている。このプロトコルを使って、リモートの計算機上で動作しているプロセスの手続きを実行するための RPC (Remote Procedure Call) という機構がある。これは、クライアントプロセスが、ある計算機上のサーバプロセスにパラメータを送り、サーバプロセスから結果を得るための手続きである。このとき、クライアントプロセスは、上述した RPC を実行するための手続き（例えば、送信データをネットワークで転送できる形式に変換したり、ポート番号を設定するなど）を行い、どの計算機上のサーバプロセスと通信するかを指定する。

【0003】 以上が従来の計算機オペレーティングシステムのネットワーク交信用基本機能として実用化されている（例えば、「UNIX ネットワークプログラミング：W. リチャード・スティーブンス著／篠田陽一訳、

トッパン」参照）。

【0004】

【発明が解決しようとする課題】 しかしながら、上述した従来の技術には、次のような課題があった。即ち、クライアントプロセスが上述した RPC を実行するとき、必ずサーバプロセスがどの計算機上で動作しているかを指定しなければならなかった。従って、クライアントプロセスが、複数のサーバプロセスの RPC を実行するときは、それぞれの RPC のサーバプロセスがどの計算機にあるかを管理しなければならないので、クライアントプロセスの構造が複雑なものとなってしまった。また、クライアントプロセスがサーバプロセスと異なる計算機上にあるときは、RPC を使い、サーバプロセスが同じ計算機上にあるときは RPC とは別のコストのかからない方法を使って通信したいときは、それぞれの通信方法でプロトコルが異なるので、統一的に扱うことができなかった。例えば、リモートの計算機上にあるサーバプロセスをローカルの計算機上に移動したとき、クライアントプログラムを大幅に書き換えなくてはならなかった。

【0005】

【課題を解決するための手段】 本発明の分散オブジェクト環境実現システムは、上述した課題を解決するため、ネットワークに接続された複数の計算機上で動作する分散オブジェクト及び当該分散オブジェクトをサーバとするクライアントの相互間の通信を制御するクライアントサーバシステムにおいて、以下の点を特徴とするものである。

【0006】 (1)

1. ネットワークに接続された各計算機上に配置された各分散オブジェクトを備える。
2. 当該各分散オブジェクトとの通信のためのプロトコルと各分散オブジェクト名とを対応させて管理するネームサーバを備える。
3. 各計算機上に設けられたクライアント内に設けられ、ネームサーバに通信先のオブジェクト名を送信するインタフェース部を備える。

【0007】 (2) (1) において、各分散オブジェクト内に設けられ、操作の依頼元のクライアント名及び操作内容の履歴を蓄積するヒストリオブジェクトを備える。

(3) (1) 又は (2) において、各分散オブジェクト内に設けられ、操作の直前のデータ内容を保存し、障害時にデータの内容を操作の直前の状態に戻すためのリカバリオブジェクトを備える。

【0008】

【作用】

(1) 本発明の分散オブジェクト環境実現システムによれば、ネームサーバによってオブジェクトの管理をしており、分散オブジェクト環境が持つ分散オブジェクトとの通信のためのインターフェース部を各クライアントに

提供している。そして、各クライアントがいずれかの分散オブジェクトにデータの操作を依頼する際は、ネームサーバに対し、分散オブジェクト名を送信する。これにより、ネームサーバが分散オブジェクトとの通信のための情報をクライアントに返信し、クライアントがその情報により通信処理を行う。従って、各クライアントはネームサーバの存在位置のみを意識すればよい。この結果、各クライアントは、ネットワークに接続された計算機上で動作している分散オブジェクトと通信するとき、その分散オブジェクトがどの計算機に存在しているのか意識する必要がない。また、オペレーティングシステムで提供されている通信プロトコルも意識する必要がない。

【0009】(2)(1)において、分散オブジェクト内のヒストリオブジェクトが通信の履歴を取るようにした結果、分散オブジェクト環境におけるアプリケーションプログラムのデバッグ作業が容易になる。

(3)(1)又は(2)において、分散オブジェクト内のリカバリオブジェクトによって操作の直前のデータ内容を管理するようにした結果、操作ミス等の障害に容易に対処することができる。

【0010】

【実施例】以下、本発明の実施例を図面を参照して詳細に説明する。図1は、本発明の分散オブジェクト環境実現システムの一実施例のブロック図である。図示のシステムは、分散オブジェクト11、12、13、14、ネームサーバ20、クライアント32、33等から成る。分散オブジェクト11、12、13、14は、それぞれ計算機1、2、3、4に配置されている。ここで、分散オブジェクトとは、オブジェクト指向の概念を用いて作られたプロセスである。各分散オブジェクトの構造は、図2のようになっている。

【0011】図2に示すように、各分散オブジェクトは、サーバ101、分散オブジェクト本体102、ヒストリオブジェクト103、リカバリオブジェクト104等から成る。このうち、サーバ101は、クライアントからの要求を処理する。サーバ101は、分散オブジェクト環境で提供され、リモートアクセス用のサーバ、ローカルアクセス用のサーバの2種類が存在する。分散オブジェクトが、リモートの計算機からメッセージを受けるときは、リモートアクセス用のサーバを使い、ローカルの計算機からしかメッセージを受けないときは、ローカルアクセス用のサーバを使う。どちらのサーバもインターフェースは同じなので、サーバを切り換えて使ってもクライアントプログラムを変更する必要はない。

【0012】図3に、分散オブジェクト本体の構造を示す。分散オブジェクト本体102は、オペレーション200の集合と、データ300の集合から構成される。そして、サーバ101が受け取ったメッセージに応じて、分散オブジェクト本体が持っているオペレーションのう

ちの1つが呼び出される。オペレーションが呼ばれると、分散オブジェクトの状態を持っているデータへの操作などの処理を行う。データ300は、分散オブジェクト本体のオペレーション以外からは操作されることはない。オペレーションが呼ばれたときの処理はアプリケーションプログラマが記述する。オペレーションの処理の中から、他の分散オブジェクトへのアクセスをすることもできる。

【0013】図4に、分散オブジェクトと通信しながら、処理を進めて行くクライアントプログラムを示す。これはアプリケーションプログラマによって作成される。クライアントプログラムの構造は、図4に示すように、ネームサーバとのインターフェース部41、分散オブジェクトとのインターフェース部42、プログラム本体部43から構成される。ネームサーバとのインターフェース部41は、分散オブジェクト環境で提供される。これは、分散オブジェクトの情報を得るためにネームサーバと通信するためのものである。これによって、ネームサーバから分散オブジェクトの情報を得ることができる。

【0014】分散オブジェクトとのインターフェース部42は、分散オブジェクト環境で提供される。これは、ネームサーバから得た情報と共に利用し、分散オブジェクトと通信するためのものである。このインターフェース部は、オペレーティングシステムで提供される、RPCのような通信プロトコルインターフェースを簡単に利用できるようにするためのものである。クライアントプログラムが複数の分散オブジェクトと通信するときには、それぞれのインターフェースが必要となる。プログラム本体部43は、分散オブジェクトと通信して得た結果などを処理する部分である。これは、アプリケーションプログラマが記述する。

【0015】図5に、ヒストリオブジェクトの構成を示す。ヒストリオブジェクト103は、クライアントから来た分散オブジェクトへのメッセージの履歴を取るものである。即ち、サーバ101がヒストリオブジェクトの持つオペレーションを起動することで、サーバ101が受け取ったメッセージをヒストリオブジェクト103に格納する。

【0016】ここで、再び図1の説明に戻る。図1において、リカバリオブジェクト104は、分散オブジェクトのメソッドの実行で障害が起こったとき、分散オブジェクトを元の状態に戻すためのものである。分散オブジェクト本体から、リカバリオブジェクトのオペレーションを呼び出すことで、分散オブジェクトの操作の直前のデータの内容を保存することができる。これによって、分散オブジェクトに障害が起こったときでも、リカバリオブジェクトに格納している保存情報を分散オブジェクトに戻せば、操作前の状態を復元することができる。従って、障害による分散オブジェクト環境への影響を少な

くすることができる。

【0017】ネームサーバ20は、各分散オブジェクト11、12、13、14についての情報を格納している。ネームサーバ20は、図示のように、分散環境上でただ1つしか存在しない。このようなネームサーバ20が格納している情報は、各分散オブジェクト名21、計算機名22、ポート番号23である。分散オブジェクト名21は、その分散オブジェクトに固有の名前である。計算機名22は、分散オブジェクトが存在している計算機の固有の名称、機器番号等である。ポート番号23は、分散オブジェクトと通信する際に指定するポート番号である。

【0018】次に、上述したシステムの動作を説明する。上述したの分散環境では、クライアントと、分散オブジェクトが通信してデータ処理を進める。クライアント32がネットワーク10で接続された計算機上で動作している分散オブジェクト11、12等と通信するために、まず、ネームサーバ20に問い合わせる。ネームサーバ20にクライアント32から要求がくると、ネームサーバ20はそれに対応する分散オブジェクトの情報を返却する。即ち、ネームサーバは分散オブジェクトの情報を持っており、それをクライアント32に返す。クライアント32はその情報を基にして、分散オブジェクト11、12等と通信できる。

【0019】分散オブジェクト11、12等は、クライアントプログラムからの要求に応じて、それに対応する処理を行う。即ち、分散オブジェクト11、12等は、上述したように、データの集合とオペレーションの集合から構成され、データへの操作はオペレーションのみによって行われる。これにより、分散オブジェクトがメッセージを受け取ると、それに対応するオペレーションを起動する。分散オブジェクト11、12等内では、クライアント32からのメッセージは、サーバ101によって次のように処理される。まず、クライアント32からのメッセージの履歴を取るために、クライアント32からのメッセージをヒストリオブジェクト103に格納する。メッセージを格納するときにはヒストリオブジェクト103の持っているオペレーションを呼ぶ。デバッグ時は、このヒストリオブジェクト103を使うことで、この分散オブジェクトを使うクライアントプログラムのデバッグを簡単にすることができる。

【0020】次に、クライアント32からのメッセージに対応する、分散オブジェクト本体102の持っているオペレーションを呼ぶ。これにより、分散オブジェクトの持つオペレーションの1つを起動する。オペレーションが起動されると分散オブジェクト本体は次のような処理をする。まず、分散オブジェクト本体102のオペレーションの処理をする前に、分散オブジェクトに障害が起こったとき、分散オブジェクトの状態を元に戻すための、分散オブジェクト本体102の状態情報をリカバリ

オブジェクト104に格納する。このとき、分散オブジェクト本体102の状態を格納するときには、リカバリオブジェクト104の持っているオペレーションを呼ぶ。その後、分散オブジェクト本体102のオペレーションを実行する。そして、オペレーションの実行後は、実行結果の回答をクライアントに返信する。

【0021】このようにして、各クライアント32又は33は、ネットワーク10に接続された計算機1、2等の上で動作している分散オブジェクト11、12等と通信するとき、その分散オブジェクト11、12等がどの計算機に存在しているのか意識する必要がなく、オペレーティングシステムで提供されている通信プロトコルも意識する必要がないようにすることができる。また、分散オブジェクト内のヒストリオブジェクト103によって通信の履歴を取るようにしたので、分散オブジェクト環境におけるアプリケーションプログラムのデバッグ作業が容易になる。更に、分散オブジェクト内のリカバリオブジェクト104によって操作の直前のデータ内容を管理するようにしたので、操作ミス等の障害に容易に対処することができる。

【0022】尚、本発明は上述した実施例に限定されるものではなく、種々の変形が可能であることはもちろんである。例えば、上述した実施例では、ヒストリオブジェクト103とリカバリオブジェクト104の双方を備えた場合について説明したが、本発明はこれに限らず、これらのうち、いずれか一方又は双方を備えない場合についても適用される。また、上述した実施例では、ネームサーバ20が単に通信情報を提供する場合について説明したが、ネームサーバ20が各分散オブジェクト11、12等への通信処理を代行するようにしてもよい。

【0023】

【発明の効果】以上説明したように、本発明の分散オブジェクト環境実現システムによれば、ネームサーバによってオブジェクトの管理をしており、分散オブジェクト環境が持つ分散オブジェクトとの通信のためのインターフェースを提供するようにしたので、次のような効果がある。即ち、クライアントがネットワークに接続された計算機上で動作している分散オブジェクトと通信するとき、そのオブジェクトがどの計算機に存在しているのか意識する必要がないようにできる。従って、クライアントプロセスの構造を単純化することができる。また、クライアントが上述した通信をするとき、オペレーティングシステムで提供されている通信プロトコルも意識する必要がないようにできる。従って、各クライアントで通信を統一的に扱うことができる。これにより、例えば、リモートの計算機上にあるサーバプロセスをローカルの計算機上に移動したときでも、クライアントプログラムを書き換える必要がないようにできる。

【図面の簡単な説明】

【図1】本発明の分散オブジェクト環境実現システムの

一実施例のブロック図である。

【図2】 分散オブジェクトの構成の説明図である。

【図3】 分散オブジェクト本体の説明図である。

【図4】 クライアントの説明図である。

【図5】 ヒストリオブジェクトの説明図である。

【符号の説明】

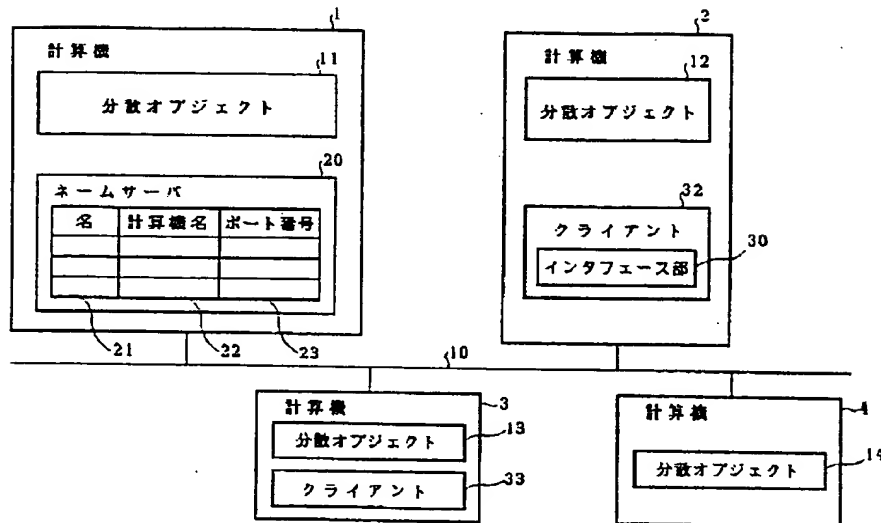
1、2、3、4 計算機

11、12、13、14 分散オブジェクト

20 ネームサーバ

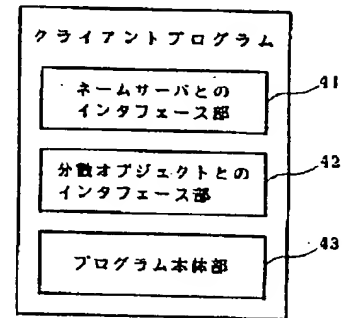
30 インタフェース部

【図1】



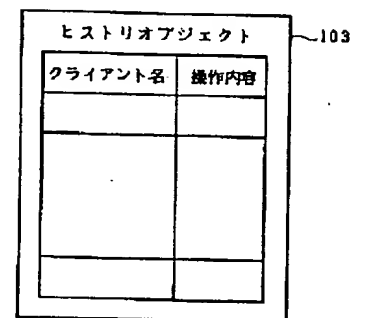
本発明の分散オブジェクト環境実現システムの一実施例

【図4】



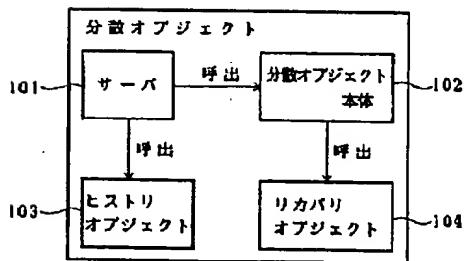
クライアントの構成

【図5】



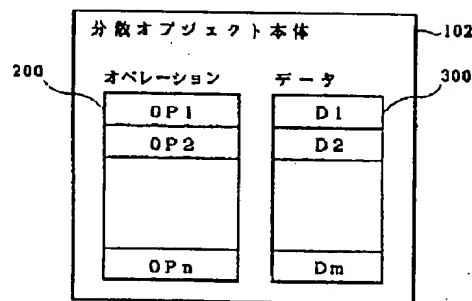
ヒストリオブジェクトの構成

【図2】



分散オブジェクトの構成

【図3】



分散オブジェクト本体の構成

This Page Blank (uspto)